<u>REMARKS</u>

Claims 1-20 and 22-24 are currently pending in the subject application and are presently under consideration. Claims 1, 15, 17, 19 and 22-23 have been amended as shown on pp. 2 and 4-6 of the Reply. The amendments to the independent claims incorporate limitations recited in claims previously presented (independent claim 24) as conveyed to the Examiner over the telephone on May 21, 2008. Accordingly, it is submitted that no new search is required and it is respectfully requested that the amendments be entered. Applicants' representative believes that the present application is now in condition for allowance, which prompt and favorable action is respectfully requested.

Favorable reconsideration of the subject patent application is respectfully requested in view of the comments and amendments herein.

I.   <u>Rejection of Claims 19 and 20 Under 35 U.S.C. §102(e)</u>

Claims 19 and 20 stand rejected under 35 U.S.C. §102(e) as being anticipated by Necula, *et al.* (US Patent 6,128,774). This rejection should be withdrawn for at least the following reasons. Necula, *et al.* does not disclose or suggest each and every aspect set forth in the subject claims.

> A single prior art reference anticipates a patent claim only if it *expressly or inherently describes each and every limitation set forth in the patent claim*. *Trintec Industries, Inc. v. Top-U.S.A. Corp.,* 295 F.3d 1292, 63 USPQ2d 1597 (Fed. Cir. 2002); *See Verdegaal Bros. v. Union Oil Co. of California,* 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). The *identical invention must be shown in as complete detail as is contained in the ... claim*. *Richardson v. Suzuki Motor Co.,* 868 F.2d 1226, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989) (emphasis added).

Applicants' claimed subject matter relates to a system and method employing plug-in pre- and/or postconditions facilitating static checking of object code. In particular, independent claims 19, as amended, recites *performing a component-wise comparison of a user injected custom state and a state defined by a parameter to determine a fault condition.* Necula, *et al.* does not disclose or suggest this novel aspect.

Necula, *et al.* relates to a system that verifies safe execution of software. In particular, the system, verifies that untrusted software supplied by a code producer is safe to execute by a

code consumer. Specifically, the code consumer can declare a precondition, stored in the configuration data, and guarantee that the precondition holds when the untrusted code is invoked. (*See* column 7, lines 19-22). Necula, *et al.* is silent with respect to a user injected custom state that can be passed to the plug-in condition to determine a fault condition and/or does not teach or suggest determination of a fault based on a comparison between a user injected custom state and a state defined by a parameter.

Applicants' claimed subject matter, in contrast, discloses a system wherein a user injected custom state is passed to a plug-in condition to determine a fault condition. In particular, the programmer/specifier can place annotation(s) in a program to be checked and interfaces invoked by the system to be checked that specify plugin-post condition(s). These post condition(s) have the effect of injecting the custom state of the programmer into the property(ies) the checker traces and propagates. Specifically, the custom state is implemented as a class whose base class is CustomStateAttribute, which the checker provides. This class has a virtual method Matches which most custom states will inherit without overriding, generally. The call S_1, Matches(S_2) tests whether the custom state S_1 is acceptable in a context that expects custom state S_2. The default implementation of this method does a component-wise comparison of the two states, allowing any value for a component whose expected value is "unknown." (*See* page 33, lines 25-31). Necula, *et al.* fails to teach or suggest these novel aspects.

In view of at least the foregoing, it is readily apparent that Necula, *et al.* does not anticipate or suggest the subject invention as recited in claim 19 (and claim 20 that depends therefrom). Accordingly, it is respectfully requested that this rejection be withdrawn.

II.   Rejection of Claims 1-6, 8-13, 15, 16 and 22-24 Under 35 U.S.C. §103(a)

Claims 1-6, 8-13, 15, 16 and 22-24 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Necula, *et al.* in view of Hallem, *et al.*("A System and Language for Building System-Specific, Static Analyses"). This rejection should be withdrawn for at least the following reasons. Necula, *et al.* either alone or in combination with Hallem *et al.,* does not teach or suggest every feature of the subject claims.

> *[T]he prior art reference (or references when combined) must teach or suggest all the claim limitations. See* MPEP § 706.02(j).
> *See also KSR Int'l Co. v. Teleflex, Inc.*, 550 U. S. ____, 04-1350,

slip op. at 14 (2007). The teaching or suggestion to make the claimed combination and the reasonable expectation of success must be found in the prior art and not based on applicant's disclosure. *See In re Vaeck,* 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991) (emphasis added).

Applicants' claimed subject matter relates to a system and method that employs pre- and/or post- condition(s) specified at a source code level and persisted (*e.g.,* in associated object code and/or a specification repository), which facilitate static checking of the object code. In particular, independent claims 1, 15 and 22-24, recite similar aspects, namely, ***performing a component-wise comparison of a user injected custom state and a state defined by a parameter to determine a fault condition***. Necula, *et al.,* alone or in combination with Hallem *et al.,* fails to teach this novel aspect.

As discussed *supra*, Necula, *et al.* does not teach or suggest a user injected custom state that can be passed to a plug-in condition and/or does not teach or suggest determination of a fault based on a comparison between a user injected custom state and a state defined by a parameter. Hallem *et al.,* fails to make up for the aforementioned deficiencies of Necula, *et al.* Hallem *et al.* merely relates to a system and language for building system-specific static analysis. The language allows users to specify a broad class of analyses in terms that resemble the intuitive description of the rules they check. However, Hallem *et al.* does not teach or suggest performing a component-wise comparison of a user injected custom state and a state defined by a parameter to determine a fault condition.

In view of at least the foregoing, it is clear that Necula, *et al.* alone or in combination with Hallem, *et al.* fails to teach or suggest all features of applicants' invention as recited in independent claims 1, 15 and 22-24 (and associated dependent claims), and thus fails to make obvious the subject claims. Accordingly, it is respectfully requested that this rejection be withdrawn.

III.   Rejection of Claim 7 Under 35 U.S.C. §103(a)

Claim 7 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Necula, *et al.* in view of Hallem, *et al.* as applied to claim 1 above, and further in view of Meijer, *et al.* ("Technical Overview of the Common Language Runtime"). Necula, *et al.* and Hallem, *et al.*,

either alone or in combination with Meijer, *et al.,* do not teach or suggest every feature of the subject claims.

Claim 7 depends on independent claim 1. As discussed above, Necula, *et al.,* alone or in combination with Hallem, *et al.* does not teach each and every aspect of independent claim 1. Necula, *et al.* and/or Hallem, *et al.* is silent with regard to determination of a fault condition by performing a component-wise comparison of a user injected custom state and a state defined by a parameter. Meijer, *et al.* discloses a Common Language Runtime (CLR) that is expressed in the Common Intermediate Language (CIL), can be compiled from a language such as C, Pascal, C# etc. However, Meijer, *et al.* does not remedy the aforementioned deficiencies presented by Necula *et al.* and/or Hallem, *et al.* with respect to the independent claim 1. Therefore, withdrawal of this rejection is respectfully requested.

IV.   Rejection of Claim 14 Under 35 U.S.C. §103(a)

Claim 14 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Necula, *et al.* in view of Hallem, *et al.* as applied to claim 1 above, and further in view of Goldberg, *et al.* (US Patent 6,571,232). This rejection should be withdrawn for at least the following reasons. The cited references do not teach or suggest all limitations recited in the subject claims.

Claim 14 depends on independent claim 1. As discussed above, Necula, *et al.* and/or Hallem, *et al.* do not teach or suggest all limitations of claim 1 and Goldberg, *et al.* is silent with regard to the aforementioned deficiencies presented by Necula, *et al.* and/or Hallem, *et al.* with respect to independent claim 1. Goldberg, *et al.* merely relates to a query object generator tool that generates interface definitions and code that implement a query object also generates a database schema access query object that retrieves the schema of an underlying database and fails to make up for the deficiencies presented by Necula, *et al.* and/or Hallem, *et al.* with respect to independent claim 1 discussed supra. Accordingly, it is respectfully requested that this rejection be withdrawn.

V.   Rejection of Claims 17 and 18 Under 35 U.S.C. §103(a)

Claims 17 and 18 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Tichelaar, *et al.* ("A Metal-model for Language-Independent Refactoring") in view of Hallem, *et al.* Tichelaar, *et al.,* alone or in combination with Hallem, *et al.* does not disclose or suggest

every feature recited in the subject claims. In particular, independent claim 17 has been amended to recite a method that performs a component-wise comparison of a user injected custom state and a state defined by a parameter to determine a fault condition. Tichelaar, *et al.* and/or Hallem, *et al.* are silent with respect to this novel aspect.

As discussed above, Hallem, *et al.* does not teach or suggest determination of a fault condition by performing a component-wise comparison of a user injected custom state and a state defined by a parameter. Tichelaar, *et al.* simply relates to a language independent refactory engine that changes a system to improve its internal structure without altering its external behavior. The engine provides standard method for programmers and tools to perform refectories no matter which language they work in. However, Tichelaar, *et al.* fails to cure the deficiencies of Hallem, *et al.* with respect to independent claim 17. Thus, it is readily apparent that Tichelaar and Hallem, *et al.*, alone or in combination, do not anticipate or suggest the subject invention as recited in claim 17 (and claim 18 that depends therefrom). Accordingly, it is respectfully requested that this rejection be withdrawn.

CONCLUSION

The present application is believed to be in condition for allowance in view of the above amendments. A prompt action to such end is earnestly solicited.

In the event any fees are due in connection with this document, the Commissioner is authorized to charge those fees to Deposit Account No. 50-1063 [MSFTP482US].

Should the Examiner believe a telephone interview would be helpful to expedite favorable prosecution, the Examiner is invited to contact applicant's undersigned representative at the telephone number listed below.

Respectfully submitted,

AMIN, TUROCY & CALVIN, LLP

/Himanshu S. Amin/
Himanshu S. Amin
Reg. No. 40,894

AMIN, TUROCY & CALVIN, LLP
24TH Floor, National City Center
1900 E. 9TH Street
Cleveland, Ohio 44114
Telephone (216) 696-8730
Facsimile (216) 696-8731